09-28-00

A

Please type a plus sign (+) inside this box → [+]

PTO/SB/05 (4/98)
Approved for use through 09/30/2000. OMB 0651-0032
Patent and Trademark Office: U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

UTILITY PATENT APPLICATION TRANSMITTAL

(Only for new nonprovisional applications under 37 C.F.R. § 1.53(b))

Attorney Docket No.	MS150832.2
First Inventor or Application Identifier	Michael Ginsberg
Title	TRUST LEVEL BASED PLATFORM ACCESS...
Express Mail Label No.	EF142130596US

APPLICATION ELEMENTS

See MPEP chapter 600 concerning utility patent application contents.

ADDRESS TO: Assistant Commissioner for Patents
Box Patent Application
Washington, DC 20231

1. [✓] * Fee Transmittal Form *(e.g., PTO/SB/17)*
(Submit an original and a duplicate for fee processing)

2. [✓] Specification [Total Pages 14]
(preferred arrangement set forth below)
- Descriptive title of the Invention
- Cross References to Related Applications
- Statement Regarding Fed sponsored R & D
- Reference to Microfiche Appendix
- Background of the Invention
- Brief Summary of the Invention
- Brief Description of the Drawings *(if filed)*
- Detailed Description
- Claim(s)
- Abstract of the Disclosure

3. [✓] Drawing(s) *(35 U.S.C. 113)* [Total Sheets 8]

4. Oath or Declaration [Total Pages]
a. [✓] Newly executed (original or copy)
b. [ ] Copy from a prior application (37 C.F.R. § 1.63(d))
(for continuation/divisional with Box 16 completed)
i. [ ] DELETION OF INVENTOR(S)
Signed statement attached deleting inventor(s) named in the prior application, see 37 C.F.R. §§ 1.63(d)(2) and 1.33(b).

5. [ ] Microfiche Computer Program *(Appendix)*

6. Nucleotide and/or Amino Acid Sequence Submission
(if applicable, all necessary)
a. [ ] Computer Readable Copy
b. [ ] Paper Copy (identical to computer copy)
c. [ ] Statement verifying identity of above copies

*** NOTE FOR ITEMS 1 & 13: IN ORDER TO BE ENTITLED TO PAY SMALL ENTITY FEES, A SMALL ENTITY STATEMENT IS REQUIRED (37 C.F.R. § 1.27), EXCEPT IF ONE FILED IN A PRIOR APPLICATION IS RELIED UPON (37 C.F.R. § 1.28).**

ACCOMPANYING APPLICATION PARTS

7. [✓] Assignment Papers (cover sheet & document(s))
8. [ ] 37 C.F.R.§3.73(b) Statement *(when there is an assignee)* [ ] Power of Attorney
9. [ ] English Translation Document *(if applicable)*
10. [ ] Information Disclosure Statement (IDS)/PTO-1449 [ ] Copies of IDS Citations
11. [ ] Preliminary Amendment
12. [✓] Return Receipt Postcard (MPEP 503)
(Should be specifically itemized)
13. [ ] * Small Entity Statement(s) *(PTO/SB/09-12)* [ ] Statement filed in prior application, Status still proper and desired
14. [ ] Certified Copy of Priority Document(s)
(if foreign priority is claimed)
15. [✓] Other: Express Mail Certificate

16. **If a CONTINUING APPLICATION,** *check appropriate box, and supply the requisite information below and in a preliminary amendment:*
[ ] Continuation [ ] Divisional [ ] Continuation-in-part (CIP) of prior application No: _____/_____
Prior application information: Examiner _____ Group / Art Unit: _____

For CONTINUATION or DIVISIONAL APPS only: The entire disclosure of the prior application, from which an oath or declaration is supplied under Box 4b, is considered a part of the disclosure of the accompanying continuation or divisional application and is hereby incorporated by reference. The incorporation can only be relied upon when a portion has been inadvertently omitted from the submitted application parts.

17. CORRESPONDENCE ADDRESS

[ ] *Customer Number or Bar Code Label* *(Insert Customer No. or Attach bar code label here)* or [✓] *Correspondence address below*

Name	Amin, Eschweiler & Turocy, LLP				
	Himanshu S. Amin				
Address	24th Floor, National City Center				
	1900 East 9th Street				
City	Cleveland	State	Ohio	Zip Code	44114
Country		Telephone	216-696-8730	Fax	216-696-8731

Name (Print/Type)	Himanshu S. Amin	Registration No. (Attorney/Agent)	40,894
Signature		Date	9/27/00

Burden Hour Statement: This form is estimated to take 0.2 hours to complete. Time will vary depending upon the needs of the individual case. Any comments on the amount of time you are required to complete this form should be sent to the Chief Information Officer, Patent and Trademark Office, Washington, DC 20231. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Assistant Commissioner for Patents, Box Patent Application, Washington, DC 20231.

PTO/SB/17 (2/98)
Approved for use through 9/30/2000. OMB 0651-0032
Patent and Trademark Office: U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

FEE TRANSMITTAL

Patent fees are subject to annual revision on October 1.
These are the fees effective November 10, 1998.
Small Entity payments must be supported by a small entity statement, otherwise large entity fees must be paid. See Forms PTO/SB/09-12.
See 37 C.F.R. §§ 1.27 and 1.28.

TOTAL AMOUNT OF PAYMENT ($) 730.00

Complete if Known	
Application Number	
Filing Date	Herewith
First Named Inventor	Michael Ginsberg
Examiner Name	
Group / Art Unit	
Attorney Docket No.	MS150832.2

METHOD OF PAYMENT (check one)

1. ☑ The Commissioner is hereby authorized to charge indicated fees and credit any over payments to:

Deposit Account Number: 50-1063
Deposit Account Name: Amin, Eschweiler & Turocy

☑ Charge Any Additional Fee Required Under 37 C.F.R. §§ 1.16 and 1.17
☐ Charge the Issue Fee Set in 37 C.F.R. § 1.18 at the Mailing of the Notice of Allowance

2. ☑ **Payment Enclosed:** ☑ Check ☐ Money Order ☐ Other

FEE CALCULATION

1. BASIC FILING FEE

Large Entity Fee Code	Large Entity Fee ($)	Small Entity Fee Code	Small Entity Fee ($)	Fee Description	Fee Paid
101	690	201	345	Utility filing fee	690
106	310	206	155	Design filing fee	
107	480	207	240	Plant filing fee	
108	690	208	345	Reissue filing fee	
114	150	214	75	Provisional filing fee	

SUBTOTAL (1) ($) 690.00

2. EXTRA CLAIM FEES

		Extra Claims		Fee from below	Fee Paid
Total Claims	20	-20** = 0	X	0	= 0
Independent Claims	3	- 3** = 0	X	0	= 0
Multiple Dependent				0	= 0

***or number previously paid, if greater; For Reissues, see below*

Large Entity Fee Code	Large Entity Fee ($)	Small Entity Fee Code	Small Entity Fee ($)	Fee Description
103	18	203	9	Claims in excess of 20
102	78	202	39	Independent claims in excess of 3
104	260	204	130	Multiple dependent claim, if not paid
109	78	209	39	** Reissue independent claims over original patent
110	18	210	9	** Reissue claims in excess of 20 and over original patent

SUBTOTAL (2) ($) -0-

FEE CALCULATION (continued)

3. ADDITIONAL FEES

Large Entity Fee Code	Large Entity Fee ($)	Small Entity Fee Code	Small Entity Fee ($)	Fee Description	Fee Paid
105	130	205	65	Surcharge - late filing fee or oath	
127	50	227	25	Surcharge - late provisional filing fee or cover sheet.	
139	130	139	130	Non-English specification	
147	2,520	147	2,520	For filing a request for reexamination	
112	920*	112	920*	Requesting publication of SIR prior to Examiner action	
113	1,840*	113	1,840*	Requesting publication of SIR after Examiner action	
115	110	215	55	Extension for reply within first month	
116	380	216	190	Extension for reply within second month	
117	870	217	435	Extension for reply within third month	
118	1,360	218	680	Extension for reply within fourth month	
128	1,850	228	925	Extension for reply within fifth month	
119	300	219	150	Notice of Appeal	
120	300	220	150	Filing a brief in support of an appeal	
121	260	221	130	Request for oral hearing	
138	1,510	138	1,510	Petition to institute a public use proceeding	
140	110	240	55	Petition to revive - unavoidable	
141	1,210	241	605	Petition to revive - unintentional	
142	1,210	242	605	Utility issue fee (or reissue)	
143	430	243	215	Design issue fee	
144	580	244	290	Plant issue fee	
122	130	122	130	Petitions to the Commissioner	
123	50	123	50	Petitions related to provisional applications	
126	240	126	240	Submission of Information Disclosure Stmt	
581	40	581	40	Recording each patent assignment per property (times number of properties)	40.00
146	760	246	380	Filing a submission after final rejection (37 CFR 1.129(a))	
149	760	249	380	For each additional invention to be examined (37 CFR 1.129(b))	

Other fee (specify) _____

Other fee (specify) _____

* Reduced by Basic Filing Fee Paid

SUBTOTAL (3) ($) 40.00

SUBMITTED BY				Complete (if applicable)	
Typed or Printed Name	Himanshu S. Amin			Reg. Number	40,894
Signature		Date	9/27/00	Deposit Account User ID	50-1063

Burden Hour Statement: This form is estimated to take 0.2 hours to complete. Time will vary depending upon the needs of the individual case. Any comments on the amount of time you are required to complete this form should be sent to the Chief Information Officer, Patent and Trademark Office, Washington, DC 20231. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Assistant Commissioner for Patents, Washington, DC 20231.

Atty. Docket No. MS150832.2

TRUST LEVEL BASED PLATFORM ACCESS REGULATION APPLICATION

by

Michael Ginsberg

I hereby certify that the attached patent application (along with any other paper referred to as being attached or enclosed) is being deposited with the United States Postal Service on this date **September 27, 2000**, in an envelope as "Express Mail Post Office to Addressee" Mailing Label Number **EF142130596US** addressed to the: Box Patent Application, Assistant Commissioner for Patents, Washington, D.C. 20231.

Jennifer C. Safranek
(Typed or Printed Name of Person Mailing Paper)

(Signature of Person Mailing Paper)

Title: Trust Level Based Platform Access Regulation Application

Reference to Related Application

This application claims the benefit of U.S. Provisional Patent Application Serial No. 60/208,502, which was filed June 5, 2000, entitled TRUST LEVEL BASED API SERVICES.

Technical Field

The present invention relates generally to computer systems. More particularly it relates to regulating access to a computer platform *via* a trust level generator and trust level monitor.

Background of the Invention

With the growth of distributed computing it has become common for many applications to seek access to other computers. Manufacturers of distributed computing platforms may want independent software producers to create applications to run on the distributed platform. Creating applications for a distributed platform is facilitated by exposing the internals of the distributed platform to the programming community. Such an exposed platform may be referred to as an open platform.

Although the platform developer may desire an open platform, the platform developer may still desire to restrict access to the platform to trusted applications that perform desired processes with no undesired effects. Conventionally, such access has been regulated by a software application. However, such a software application may itself not be provably trustworthy by the platform. Further, conventional access regulation systems have provided only a binary solution, either granting or denying access. Further still, many conventional access regulation systems generally provided application level verification.

As distributed platforms have become smaller, it has become more common to embed certain programs in the distributed platform. Some embedded programs may be developed and tested by the distributed platform manufacturer and thus may be considered trustworthy. Other embedded programs may have been developed and tested

by third parties and thus may not be considered trustworthy. However, conventional access regulation systems may have treated such programs similarly.

Thus, there is a need for an access regulation system that is provably trustworthy, that can provide greater flexibility than a binary response and that can analyze and interact with a computing environment, rather than simply with stand alone applications.

Summary of the Invention

The present invention provides an operating system component that determines when an application desires access to a distributed platform. One method an application may use to access a platform is *via* one or more application programming interfaces (APIs). The operating system component regulates access to the platform and such regulation may be achieved *via* limiting calls that an application can make through one or more APIs. The present invention further includes a distributed platform trustworthiness analysis application for analyzing applications attempting to access a distributed platform. The analysis application establishes a trust level for the application seeking to access the distributed platform. The trust level determines which calls, if any, to one or more APIs may be permitted. The present invention further includes a component for monitoring the trust level established by the verification program for separate interpretation of the trust level of other modules called by the application that desires access to the distributed platform. The trust level monitoring program thus facilitates interaction with a program and the programming environment in which it is executed.

If a trust level is established for an application seeking access to the distributed platform, that trust level may be examined when the application calls other modules, for example dynamic link libraries. If the dynamic link library has a lower trust level than the application, the dynamic link library may not be permitted to load and thus may be denied access to the distributed platform. Thus, a trusted application may not be compromised by a less trusted library. Conversely, if a "run restricted" application calls a "fully trusted" dynamic link library, the dynamic link library may be treated as though it were "run restricted", because of its association with the "run restricted" application.

Thus, as illustrated above, the present invention mitigates the problem in conventional systems of a binary response to verification of access to distributed platforms by providing for at least three trust levels for applications. Further, the present invention also mitigates the problems associated with conventional systems concerning analyzing applications individually, without regard to the trust environment. Still further, since the operating system component and the analysis component may be embedded in ROM in the distributed platform, the operating system component and the analysis component may be verifiably trust-worthy, solving yet another problem.

To the accomplishment of the foregoing and related ends, certain illustrative aspects of the invention are described herein in connection with the following description and the annexed drawings. These aspects are indicative, however, of but a few of the various ways in which the principles of the invention may be employed and the present invention is intended to include all such aspects and their equivalents. Other advantages and novel features of the invention may become apparent from the following detailed description of the invention when considered in conjunction with the drawings.

Brief Description of the Drawings

Fig. 1 is a schematic block diagram illustrating the establishment of a trust level for a module in accordance with an aspect of the present invention;

Fig. 2 is a schematic block diagram illustrating a Read Only Memory (ROM) containing an operating system component and a verification component in accordance with an aspect of the present invention;

Fig. 3A is a schematic block diagram illustrating a module's access to a restricted area being limited via a trust level being applied to an Application Programming Interface (API);

Fig. 3B is a schematic block diagram illustrating a module's access to a distributed platform being limited *via* a trust level selectively limiting calls to an API in accordance with an aspect of the present invention;

Fig. 4 is a schematic block diagram illustrating a second module's access to a distributed platform being limited by applying the trust level established for the calling module in accordance with an aspect of the present invention;

Fig. 5 is a table illustrating the interaction between trust levels of modules in accordance with an aspect of the present invention;

Fig. 6 is a flow chart illustrating a method for regulating access to a platform in accordance with an aspect of the present invention; and

Fig. 7 is a flow chart illustrating a method for establishing a trust level for a module in accordance with an aspect of the present invention.

Detailed Description of the Invention

The present invention is now described with reference to the drawings, wherein like reference numerals are used to refer to like elements throughout. In the following description, for purposes of explanation, numerous specific details are set forth in order to provide a thorough understanding of the present invention. It may be evident, however, to one skilled in the art that the present invention may be practiced without these specific details. In other instances, well-known structures and devices are shown in block diagram form in order to facilitate description of the present invention.

Fig. 1 is a schematic block diagram illustrating an operating system component 10 for detecting when a module 12 seeks to run on a distributed platform 14. For example, the operating system component 10 may receive a signal or a command to load the module 12. If the operating system component 10 determines that the module 12 seeks to run on the distributed platform 14, then the operating system component 10 may transmit the module 12 to an analyzing component 16 that will establish and store a trust level 18 for the module 12 in accordance with an aspect of the present invention. The analyzing component 16 may establish, for example, one of three values for the trust level 18 for the module 12, such as (1) fully trusted, (2) run restricted, and (3) fail to load. For example, the analyzing component 16 may verify a checksum, or may apply an integrity algorithm to the module 12 to determine whether it should be permitted access to a restricted area 20 of the distributed platform 14. The trust level 18 may be employed to restrict the module's 12 access to the restricted area 20 in the distributed platform 14.

For example, a first trust level may allow the module 12 read and write access to the restricted area 20 while a second trust level may allow the module 12 read-only access to the restricted area 20. Allowing multiple trust levels mitigates the binary

response to verification problems. Conventionally, programs either had complete access or no access. While some programs may not be trustworthy enough to read and write the restricted area 20 of the distributed platform 14, those programs may be trustworthy enough to read the restricted area 20 and thus the multiple trust levels enable improved control of access to the distributed platform 14.

For example, a cellular telephone distributed platform may include an embedded operating system and an embedded analyzing program. The operating system may include a component 10 to determine when a module 13a is trying load onto the cell phone. Before loading the module 13a, the operating system component 10 may transmit the module 13a to the embedded analyzing component 16, which determines whether the module 13a may access the cell phone, and thus whether the module may execute, and if so, with what level of trust. The embedded analyzing component may establish, for example, one of three trust levels for the module, like (1) fully trusted, (2) run restricted, and (3) fail to load. Based on the trust level established, the module 13a, if permitted to load, may be permitted to read and write the restricted areas of the cell phone, or it may only be permitted to read the restricted areas. One such restricted area may be the registry area of the cell phone.

To determine the trust level 18, the analyzing component 16 may utilize one or more trustworthiness verification techniques well known in the art. For example, if a module 13b written by cell phone manufacturer seeks to load onto the cell phone, that program may contain an encrypted key known to the analyzing component 16 and a cyclic redundancy character generated by an algorithm known to the analyzing component 16. The module 13b may be transmitted to the analyzing component 16 that will verify the key and the cyclic redundancy character and establish a "fully trusted" trust level. Further illustrating how a trust level 18 may be established, consider another module 13c, also written by the cell phone manufacturer, that may seek to load onto the cell phone. This module 13c may have been hacked by a third party and thus either the encrypted key or the cyclic redundancy character may not be in a desired state. Thus, the analyzing component 16 may establish a "fail to load" trust level 18. Yet another module 13d, written by a third party, may also seek to load onto the cell phone. The analyzing component 16 may scan the module 13d for viruses or other code that would make the

module 13d not trustworthy. After establishing that the module 13d is not going to compromise the cell phone, the analyzing component 16 may establish a "run restricted" trust level to allow the application to run on the cell phone but not to allow it to alter the internals of the cell phone. Thus, third party applications may be written for the cell phone without compromising the cell phone security, based upon the methods for establishing a trust level 18 described above.

Fig. 2 is a schematic block diagram illustrating a Read Only Memory (ROM) 40 containing an operating system component 42 and an analyzing component 44. By the manufacturer of the distributed platform 14 embedding the operating system component 42 and the analyzing component 44 in the ROM 40, both the operating system component 42 and the analyzing component 44 may be treated as trustworthy by the distributed platform 14. Conventionally, the operating system component 42 was stored in Random Access Memory (RAM), which is vulnerable to corruption. Similarly, the analyzing component 44 was conventionally stored in RAM, similarly subject to corruption. Because the operating system component 42 and the analyzing component 44 were subject to corruption, they were not verifiably trustworthy.

The ROM 40 may also contain a modules section 46 and a files section 48. The modules section 46 may be utilized by the manufacturer of the distributed platform 14 to embed programs that have been pre-analyzed and pre-determined to be fully trustworthy. Similarly, the files section 48 may be utilized by the manufacturer of the distributed platform 14 to embed programs that have not been pre-analyzed and pre-determined to be fully trustworthy yet which the manufacturer desires to have embedded in the distributed platform. Programs placed in the modules section 46 may not be transmitted by the operating system component 42 to the analyzing component 44 as they may be treated as trustworthy by the distributed platform 14. Such programs may automatically have their trust level 50 set to "fully trusted", for example. Similarly, the ROM 40 may contain the files section 48 which may also contain programs. But the programs in the files section may not be automatically granted a "fully trusted" trust level 50 and thus may be transmitted to the analyzing component 44 because they may not be treated as trustworthy. Embedding the operating system component 42 and the verification

component 44 in the ROM 40 mitigates the problem in the prior art of having a verification component that is not itself verifiably trustworthy.

Fig. 3A is a schematic block diagram illustrating a module 12 having access to a restricted area 52 limited by the application of a trust level 18 to an API 68. The module 12 may make one or more calls that are intended to read and or write the restricted area 52. Some calls may be blocked in the API 68 by the application of the trust level 18. But other calls may not be blocked in the API 68 by the application of the trust level 18 and may thus read and write the restricted area 52. Still other calls may be partially blocked in the API 68 by the application of the trust level 18 and thus may read but not write the restricted area 52.

Fig. 3B is a schematic block diagram illustrating an operating system component 60 limiting a module's 62 access to a distributed platform 64 by applying the trust level 18, established, for example, using the methods described in the description associated with Fig. 1, to selectively limit the module's 62 ability to make calls to an Application Programming Interface (API) 68. A plurality of calls may be directed to the API 68 from the module 62. The calls may include, for example, calls to read and/or write an area in the distributed platform 64, and to perform some logic processing and/or to perform some input/output processing on the distributed platform 64 for example. If a "fully trusted" trust level 66 was established, for example, after the analyzing component 16 (Fig. 1) verified an encrypted keyword and CRC, then all calls from the module 62 to the API 68 may be permitted to access the distributed platform 64. But if a "run restricted" trust level 66 was established, for example, after the analyzing component 16 determined that no viruses were present in a third party module 12, then some calls may selectively be blocked. For example, a call 70A for reading part of a registry 72 of the distributed platform 64 may be permitted while a call 70B for writing to the registry 72 may not be permitted based upon the determined trust level. Similarly, a call 70C may have both reading and writing components and may be partially disabled, allowing the reading functionality to process but not allowing the writing functionality to process. Allowing some calls to complete successfully, while preventing other calls from completing, facilitates multiple levels of trust, thus mitigating the binary access/no-access response problem associated with conventional systems.

Fig. 4 is a schematic block diagram illustrating an operating system component 80 limiting access to a distributed platform 82 by a second module 84 called by a first module 86 by applying a trust level 88 established for the first module 86. As discussed above, the trust level 88 may have been determined by the analyzing component 16 (Fig. 1) applying one or more well known verification algorithms and/or techniques. The first module 86 may be an application with a certain limited functionality. Thus, the first module 86 may rely on one or more second modules 84 to perform additional functionality. Both the first module 86 and the second module 84 may have their own trust level but the trust level 88 of the first module 86 is utilized to determine the relative trust level of the second module 84. Such a relative trust level may differ from the trust level that the second module 84 would have received if it had been analyzed individually. Thus, the second module 84 is analyzed not only by itself but as part of an application environment 90 that includes the context of the first module 86 and its associated trust level 88. Interaction of relative trust levels between modules is illustrated below, in Fig. 5.

Fig. 5 is a table illustrating the interaction between trust levels of an application 100 calling a Dynamic Link Library (DLL) 102 as discussed in the description accompanying Fig. 4. If the application 100 is "fully trusted", and the DLL 102 is "fully trusted", then the DLL 102 is treated as "fully trusted". But if the application 100 is "fully trusted" and the DLL 102 trust level is analyzed to be "run restricted", then the DLL 102 may not be permitted to load since its lower trust level may compromise the "fully trusted" status of the application 100. Thus, applications with higher trust levels are not corrupted by DLLs with lower trust levels. If the application 100 is "run restricted" and calls the DLL 102 that is "fully trusted", then the DLL 102 may be downgraded to "run restricted" because of its association with the application 100. Thus, a less trusted application may not be permitted greater access *via* a more trusted DLL.

Fig. 6 is a flow chart illustrating a method for regulating access to a platform. At step 110 a signal is received indicating that a module desires access to a platform. For example, the operating system may receive an interrupt indicating that a module seeks to load and/or the operating system may receive a call intended for an API. At step 112, a determination is made concerning whether the module already has a trust level

established. For example, a module seeking to load for the first time may not have a trust level established because it has not yet been analyzed for trustworthiness while a module already loaded but seeking to make a call *via* an API may have a trust level established because it has already been analyzed for trustworthiness. If the determination at step 112 is that no trust level has been established, then at step 114 the trust level is established. Step 114 is illustrated further in Fig. 7. If the determination at step 112 is that a trust level has been established, then at step 116 a determination is made to determine whether the trust level is high enough to allow the desired access. If the trust level meets or exceeds a pre-determined threshold level, then at step 118, the desired access is permitted. Such a threshold level may be, for example, "fully trusted". It is to be appreciated by one skilled in the art that different modules may have different threshold levels. For example, the module may be permitted to load, or the call to the API may be permitted to complete. If the trust level does not meet or exceed a pre-determined threshold level, then at step 120 a determination is made concerning whether the module should be terminated. If the module should be terminated, then at step 122 the module is terminated. For example, if the module was trying to write to a part of the platform that only the operating system is permitted to access, then the module may need to be terminated. If the module should not be terminated, then at step 124 the desired functionality is not permitted. For example, the module may not be loaded or the call to the API may not be permitted to complete.

Fig. 7 is a flow chart illustrating a method for establishing a trust level for a module. As discussed above, some modules may be pre-analyzed by the platform developer and thus pre-determined to be fully trustworthy. Such modules may be stored in a section of a ROM on the platform known as a modules section. At step 130 a determination is made whether the module is in the modules section of the ROM. If the module is in the modules section of the ROM, then at step 132 the trust level for the module is set to "fully trusted". Modules in the modules section of the ROM are located in that section by the manufacturer of the platform to indicate that they should be accorded "fully trusted" status. If the module is not in the modules section of the ROM, then at step 134, the module is transmitted to the analysis program to determine its trust level. At step 136, the results of the analysis are examined. If the analysis determined

that the module should not be loaded, for example if a virus was detected or if destructive code was encountered, then at step 138 a trust level of "load fail" is established, the module is not loaded and the transmitted version is deleted from the process space associated with the analysis program. If the analysis determined that the module should be permitted to load, for example, one or more of the verification algorithms and/or techniques well known in the art indicate that the module is trustworthy, then at step 140 a determination is made whether the module is fully trustworthy. If the module is fully trustworthy, then at step 142 a trust level of "fully trusted" is generated and stored in the process space associated with the module. If the module is not fully trusted, then at step 144 a trust level of "run restricted" is generated and stored in the process associated with the module. Then at step 146, the module is loaded and run, subject to the trust level established.

What has been described above includes examples of the present invention. It is, of course, not possible to describe every conceivable combination of components or methodologies for purposes of describing the present invention, but one of ordinary skill in the art may recognize that many further combinations and permutations of the present invention are possible. Accordingly, the present invention is intended to embrace all such alterations, modifications and variations that fall within the spirit and scope of the appended claims. Furthermore, to the extent that the term "includes" is used in either the detailed description or the claims, such term is intended to be inclusive in a manner similar to the term "comprising."

Claims

What is claimed is:

1. A system for regulating access to a platform comprising:

a component for analyzing a first module and an application environment associated with the first module and determining a level of access to the platform, and applying a trust level to the first module corresponding to the determined level of access.

2. The system of claim 1, the component for analyzing the first module providing for inheritance of the trust level.

3. The system of claim 1, the component for analyzing the first module providing for marking the first module with at least one of states: (1) fully trusted, (2) run restricted, and (3) fail to load.

4. The system of claim 1, wherein the component is stored in a Read Only Memory (ROM) in the platform.

5. The system of claim 1, wherein the component is part of an operating system.

6. The system of claim 1, wherein the trust level is utilized to regulate access to the platform of one or more second modules called by the first module.

7. The system of claim 1, wherein the functionality of one or more Application Programming Interface (API) calls, when called by the first module, are selectively restricted.

8. The system of claim 7, wherein selectively restricting the functionality of the one or more API calls includes restricting the functionality to read functions.

9. The system of claim 8, wherein selectively restricting the functionality of the one or more API calls includes terminating the first module.

10. A system for regulating access to a platform, comprising:
means for determining a trust level for a first module; and
means for applying the trust level to the first module to regulate access to the platform.

11. The system of claim 10 further comprising means for applying the trust level to one or more second modules called by the first module.

12. A method for regulating access to a platform, comprising the steps of:
determining a trust level for a first module; and
applying the trust level to the first module to regulate access to the platform.

13. The method of claim 12 wherein determining the trust level for the first module further comprises the step of marking the first module with at least one of states: (1) fully trusted, (2) run restricted, and (3) fail to load.

14. The method of claim 12 wherein determining the trust level for the first module further comprises transmitting the first module to a verification program.

15. The method of claim 12 wherein regulating access to the platform further comprises selectively aborting calls made to one or more APIs.

16. The method of claim 12 wherein regulating access to the platform further comprises selectively terminating the first module.

17. The method of claim 12 wherein the program for determining the trust level for the first module is stored in a ROM in the platform.

18. The method of claim 12 wherein the logic for applying the trust level to regulate access to the platform is stored in a ROM in the platform.

19. The method of claim 12 wherein the trust level may be inherited.

20. The method of claim 12 wherein the trust level may be applied to one or more second modules called by the first module.

Abstract of the Invention

The present invention provides a system and method for regulating access to a computer platform *via* a provably trustworthy trust level generator and monitor. The present invention comprises an operating system component that recognizes when applications desire access to a distributed platform. The operating system component is responsible for regulating access to the platform. Such regulation may be achieved by, for example, refusing to load the application or by limiting calls that an application can make through one or more Application Programming Interfaces. The present invention further comprises a distributed platform analysis component for analyzing applications attempting to access a distributed platform and for establishing a trust level for the application. The present invention further provides a system and method for monitoring the trust level established by the analysis program for separate interpretation of the trust level of other modules called by the application seeking access to the distributed platform.

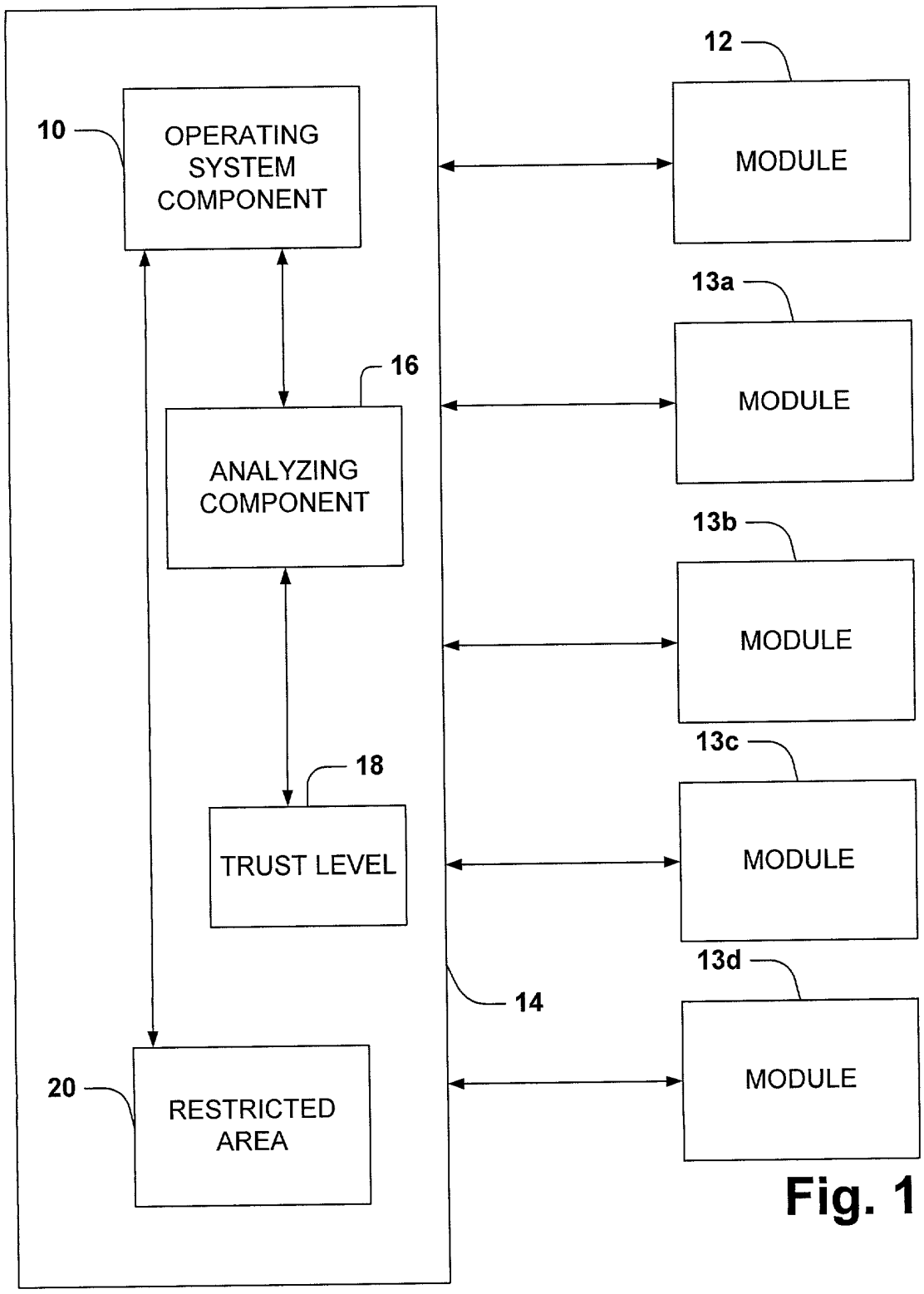
10
OPERATING SYSTEM COMPONENT
16
ANALYZING COMPONENT
18
TRUST LEVEL
20
RESTRICTED AREA
14
12
MODULE
13a
MODULE
13b
MODULE
13c
MODULE
13d
MODULE

Fig. 1

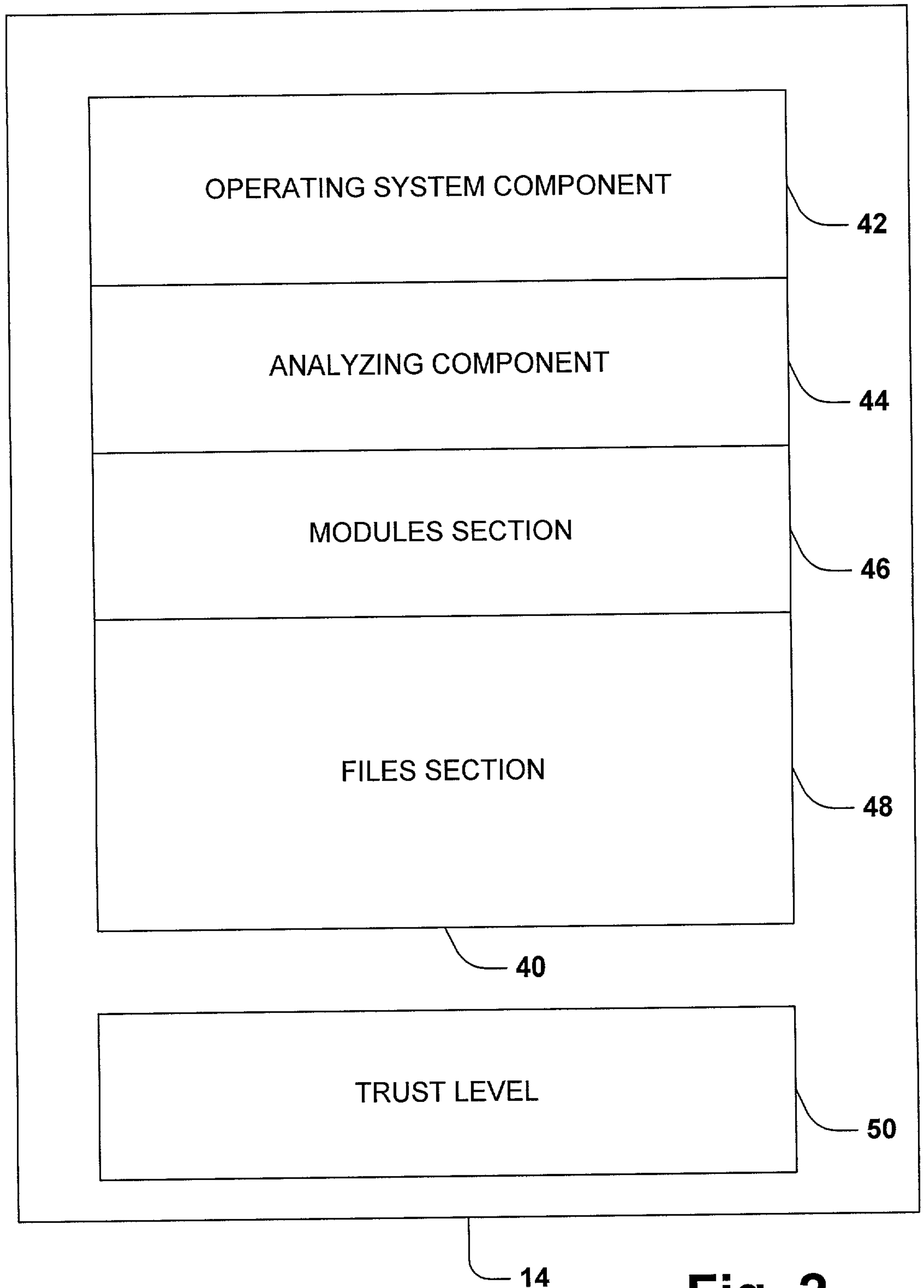
OPERATING SYSTEM COMPONENT
42
ANALYZING COMPONENT
44
MODULES SECTION
46
FILES SECTION
48
40
TRUST LEVEL
50
14

Fig. 2

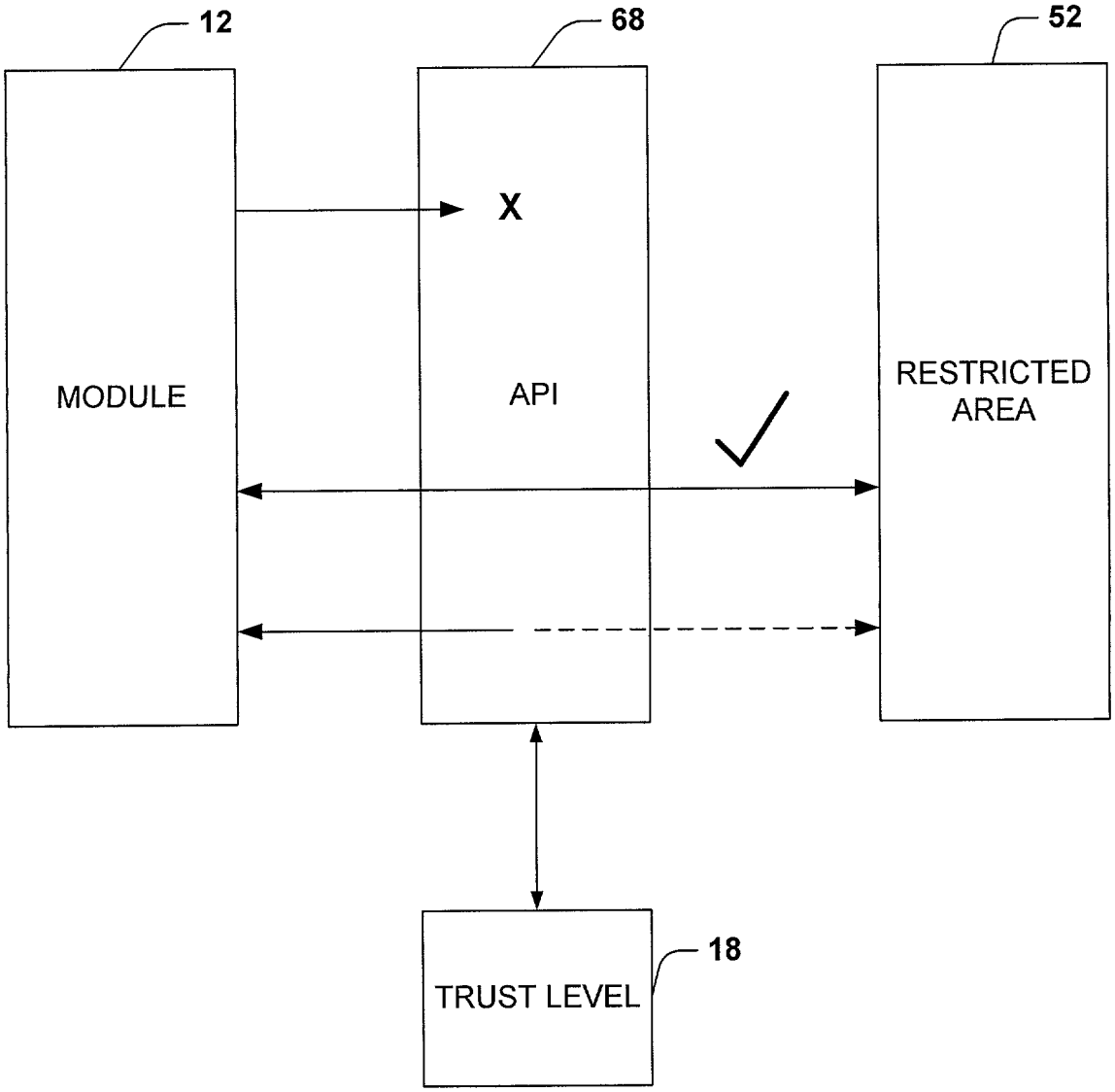

Fig. 3a

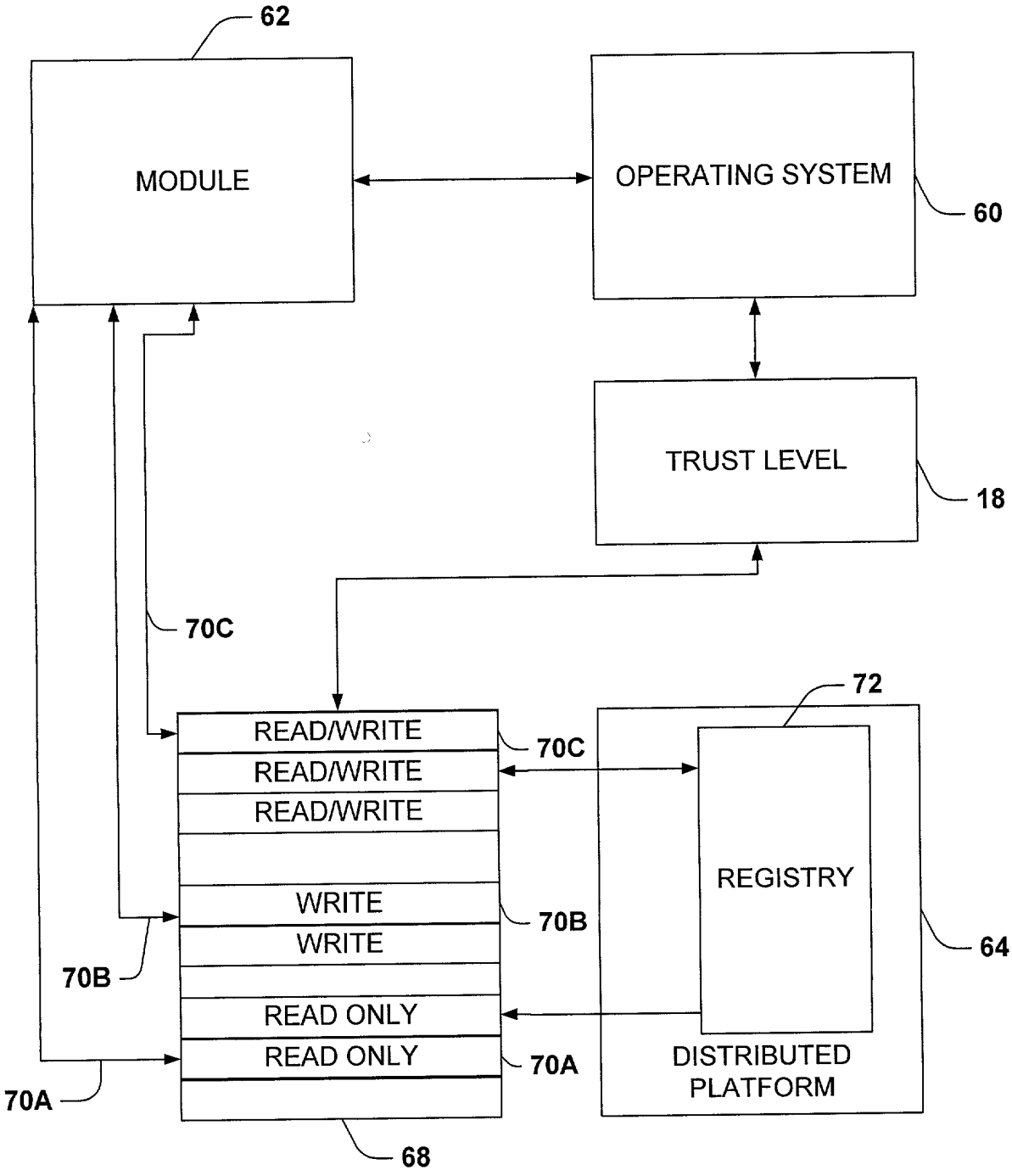
62
MODULE
OPERATING SYSTEM
60
TRUST LEVEL
18
70C
72
READ/WRITE
READ/WRITE
READ/WRITE
70C
REGISTRY
WRITE
WRITE
70B
70B
64
READ ONLY
READ ONLY
70A
DISTRIBUTED
PLATFORM
70A
68

Fig. 3b

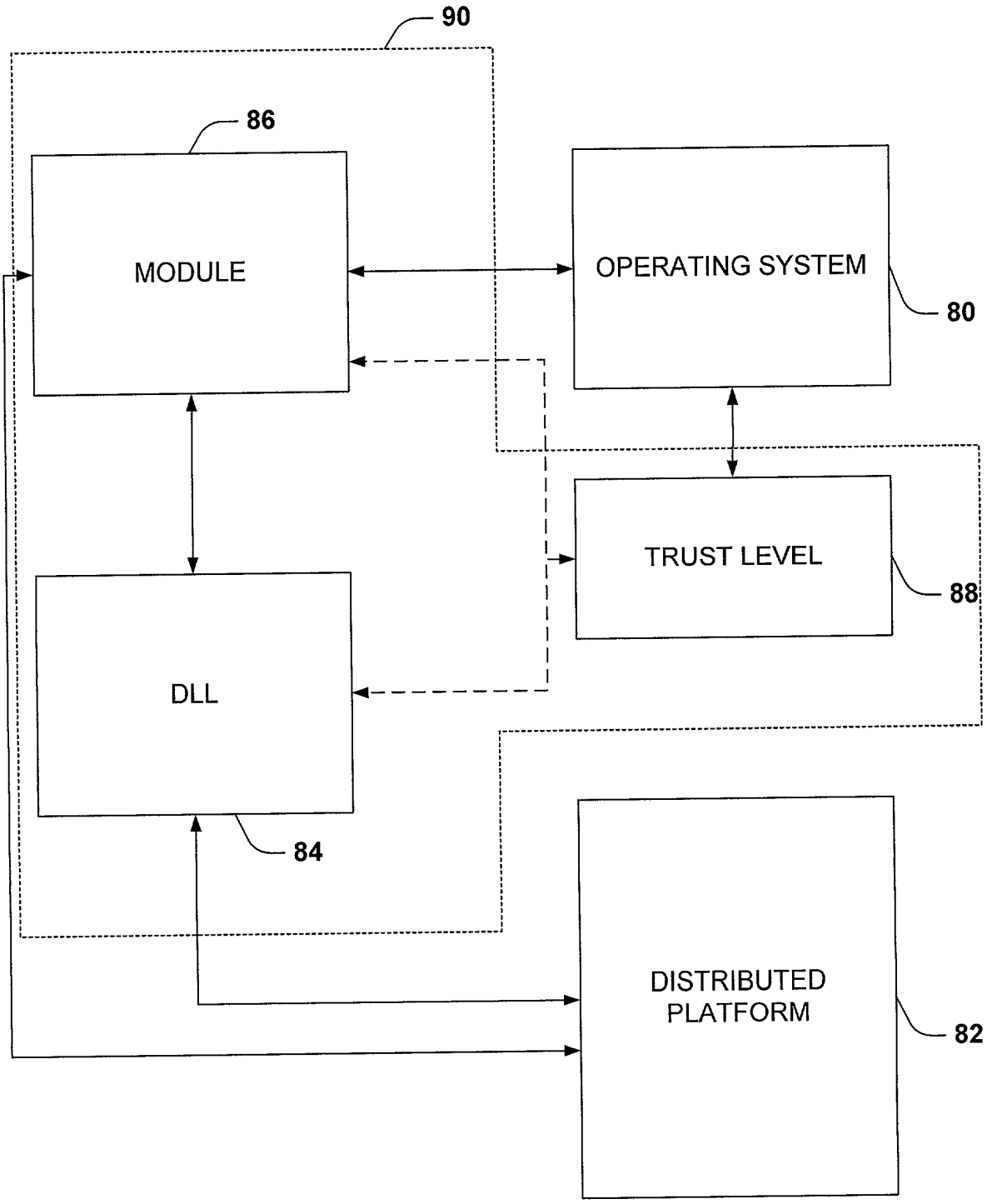
90
86
MODULE
OPERATING SYSTEM
80
TRUST LEVEL
88
DLL
84
DISTRIBUTED
PLATFORM
82

Fig. 4

100

102

		DLL		
		Fully Trusted	Run Restricted	Fail To Load
APPLICATION	Fully Trusted	Fully Trusted	Load Fails	Load Fails
	Run Restricted	Run Restricted	Run Restricted	Load Fails
	Fail To Load	Load Fails	Load Fails	Load Fails

Fig. 5

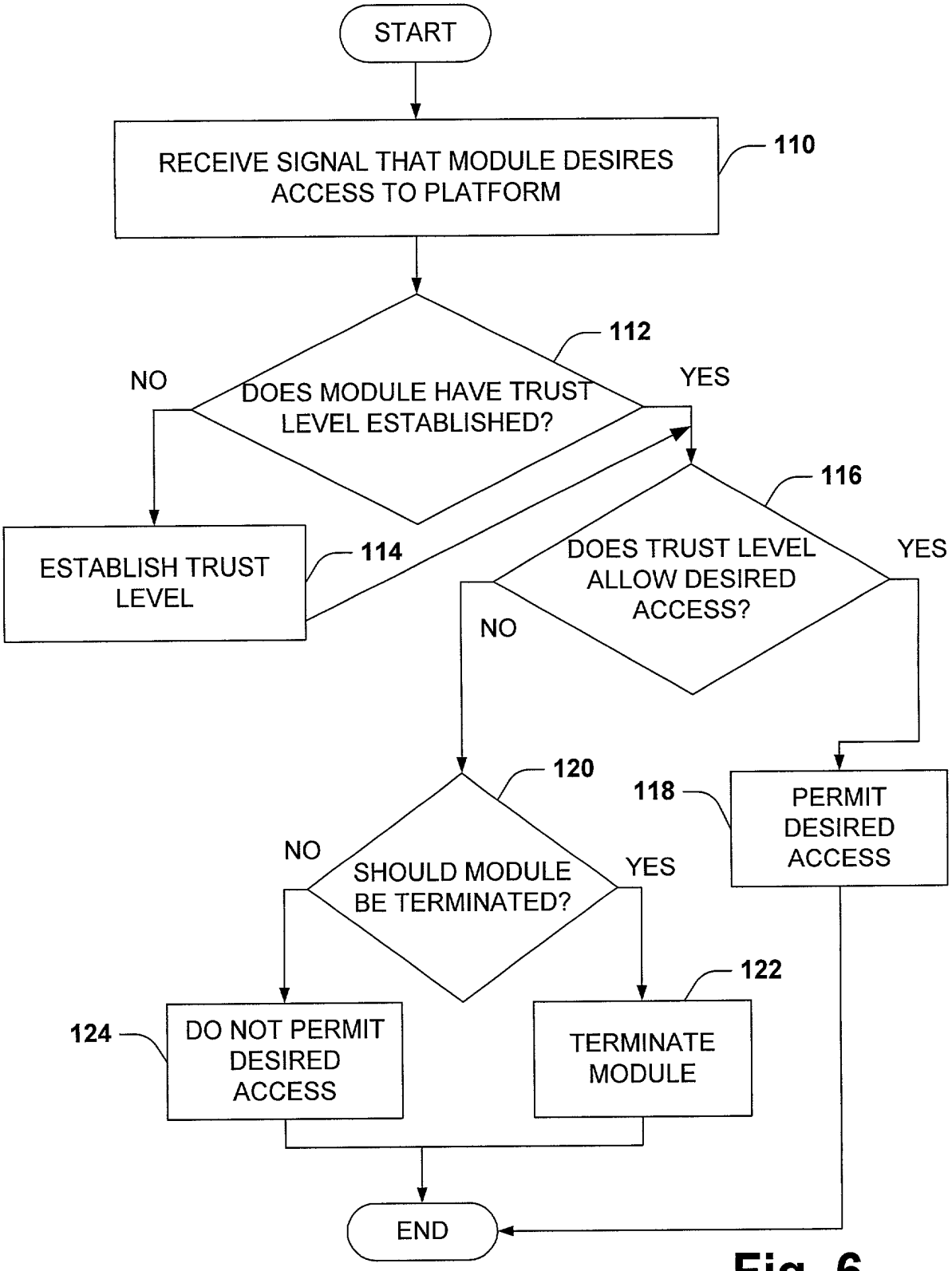
START
RECEIVE SIGNAL THAT MODULE DESIRES ACCESS TO PLATFORM
110
DOES MODULE HAVE TRUST LEVEL ESTABLISHED?
112
NO
YES
ESTABLISH TRUST LEVEL
114
DOES TRUST LEVEL ALLOW DESIRED ACCESS?
116
NO
YES
SHOULD MODULE BE TERMINATED?
120
NO
YES
PERMIT DESIRED ACCESS
118
DO NOT PERMIT DESIRED ACCESS
124
TERMINATE MODULE
122
END

Fig. 6

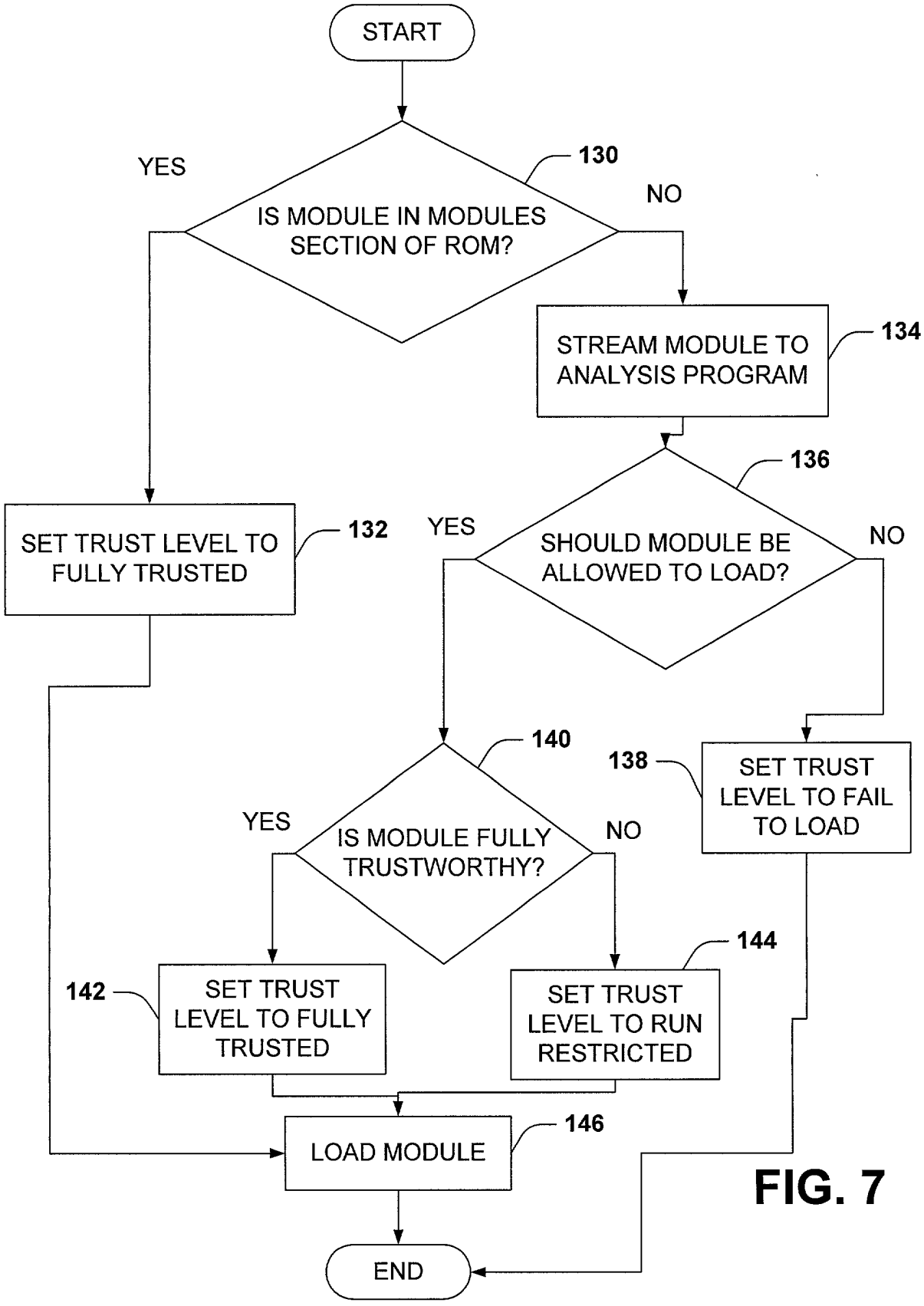
START
IS MODULE IN MODULES SECTION OF ROM?
130
YES
NO
STREAM MODULE TO ANALYSIS PROGRAM
134
SHOULD MODULE BE ALLOWED TO LOAD?
136
YES
NO
SET TRUST LEVEL TO FULLY TRUSTED
132
SET TRUST LEVEL TO FAIL TO LOAD
138
IS MODULE FULLY TRUSTWORTHY?
140
YES
NO
SET TRUST LEVEL TO FULLY TRUSTED
142
SET TRUST LEVEL TO RUN RESTRICTED
144
LOAD MODULE
146
END

FIG. 7

COMBINED DECLARATION AND POWER OF ATTORNEY
(ORIGINAL, DESIGN, NATIONAL STAGE OF PCT)

As a below named inventor, I hereby declare that:

My residence, post office address and citizenship are as stated below next to my name, I believe I am the original, first and sole inventor (if only one name is listed below) or an original, first and joint inventor (if plural names are listed below) of the subject matter which is claimed and for which a patent is sought on the invention

entitled: **TRUST LEVEL BASED PLATFORM ACCESS REGULATION APPLICATION**

the specification of which

(a) _X_ is attached hereto.

(b) ___ was filed on _____ as Serial No. 09 /_____ or
Express Mail No._____, as Serial No. not yet known, and was amended on (if applicable).

(c) ___ was described and claimed in PCT International Application No. _____ filed on _____ and amended under PCT Article 19 on _____ (if any).

I hereby state that I have reviewed and understand the contents of the above identified specification, including the claims, as amended by any amendment referred to above.

I acknowledge the duty to disclose information which is material to patentability in accordance with Title 37, Code of Federal Regulations §1.56(a).

PRIORITY CLAIM

I hereby claim priority benefits under Title 35, United States Code, §119 of any provisional application, or any foreign application(s) for patent or inventor's certificate or of any PCT international application(s) designating at least one country other than the United States of America listed below and have also identified below any foreign application(s) for patent or inventor's certificate or any PCT international application(s) designating at least one country other than the United States of America filed by me on the same subject matter having a filing date before that of the application(s) of which priority is claimed.

(d) ___ no such applications have been filed.

(e) _X_ such applications have been filed as follows.

EARLIEST FOREIGN OR PROVISIONAL APPLICATION(S), IF ANY FILED WITHIN 12 MONTHS (6 MONTHS FOR DESIGN) PRIOR TO THIS U.S. APPLICATION

COUNTRY	APPLICATION NUMBER	DATE OF FILING (day, month, year)	PRIORITY CLAIMED UNDER 35, USC 119
U.S.	60/209,502	June 5,2000	X Yes ___ No
			___ Yes ___ No
			___ Yes ___ No

ALL FOREIGN APPLICATION(S), IF ANY FILED MORE THAN 12 MONTHS (6 MONTHS FOR DESIGN) PRIOR TO THIS U.S. APPLICATION

_____

_____

POWER OF ATTORNEY

As a named inventor, I hereby appoint the following attorney(s) and/or agent(s) to prosecute this application and transact all business in the Patent and Trademark Office connected therewith. (List name and registration number)

Himanshu S. Amin, Reg. No. 40,894; Gregory Turocy, Reg. No. 36, 952;
Christopher P. Harris, Reg. No. 43,660; Eric M. Highman,
Reg. No. 43,672; and Gary J. Pitzer, Reg. No. 39,334.

Katie E. Sako, Reg. No. 32,628 and Daniel D. Crouse, Reg. No. 32,022.

The undersigned to this declaration and power of attorney hereby authorizes the U.S. attorney(s) named herein to accept and follow instructions from:

Name(s) of authorized representative(s)_____

Address_____

as to any actions to be taken in the Patent and Trademark Office regarding this application without direct communication between the U.S. attorney(s) and the undersigned. In the event of a change in the person(s) from whom instructions may be taken, the U.S. attorney(s) will be so notified by the undersigned.

Send Correspondence To:

Himanshu S. Amin
AMIN, ESCHWEILER & TUROCY, LLP
24TH Floor, National City Center
1900 East 9TH Street
Cleveland, Ohio 44114

Direct Telephone Calls To:
(name and telephone number)

Himanshu S. Amin

(216) 696-8730

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with knowledge that willful false statements and the like are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code, and that such willful false statements may jeopardize the validity of the application or any patent issued therein.

Full name of **sole or first inventor**, if any: **Michael Ginsberg**
Inventor's signature: Michael Ginsberg
Date: 9/21/00 Country of Citizenship: **U.S.**
Residence: **Redmond, Washington**
Post Office Address: **26125 NE 27th Drive**
Redmond, Washington 98053

CHECK FOR ANY OF THE FOLLOWING ADDED PAGE(S) WHICH FORM A PART OF THIS DECLARATION

X This declaration ends with this page.